Improving Deep Learning Approaches for Human Activity Recognition based on Natural Language Processing of Action Labels

1st Konstantinos Bacharidis
Computer Science Department, University of Crete, and Institute of Computer Science, FORTH
Heraklion, Greece
kbach@ics.forth.gr

2nd Antonis Argyros
Computer Science Department, University of Crete, and Institute of Computer Science, FORTH
Heraklion, Greece
argyros@ics.forth.gr

***Abstract*—Human activity recognition has always been an appealing research topic in computer vision due its theoretic interest and vast range of applications. In recent years, machine learning has dominated computer vision and human activity recognition research. Supervised learning methods and especially deep learning-based ones are considered to provide the best solutions for this task, achieving state-of-the art results. However, the performance of deep learning-based approaches depends greatly on the modelling capabilities of the spatio-temporal neural network architecture and the learning goals of the training process. Moreover, the design complexity is task-depended. In this paper, we show that we can exploit the information contained in the label description of action classes (action labels) to extract information regarding their similarity which can then be used to steer the learning process and improve the activity recognition performance. Moreover, we experimentally verify that the adopted strategy can be useful in both single and multi-stream architectures, providing better scalability on the training of the network in more complex datasets featuring activity classes with larger intra- and inter-class similarities.**

***Index Terms*—Human activity recognition, Deep learning, Natural Language Processing**

I. Introduction

Human activity recognition is a challenging problem of computer vision. It is currently undergoing rapid advancement due to the multiple and diverse applications ranging from smart home applications, to support of Human-Robot Collaboration (HRC) in complicated scenarios. Despite the achieved progress, a number of challenges remain, particularly with respect to the spatio-temporal modeling of composite activities. These challenges can be attributed to numerous factors such as the high-dimensionality of the video data, viewpoint changes, camera motion, intra-class variations and many others [1].

Robust recognition of human activities requires methods that are able to accurately distinguish underlying sub-actions and to provide strong representations for their temporal relationships and ordering. Moreover, the overall representation needs to be unique in order to be able to differentiate between activities with common sub-actions. In realistic conditions, activities can consist of actions that can potentially share similar motion patterns or appearance characteristics. An ideal approach should be able to generalize in simple but also in complex activity tasks. In recent years, deep neural network (DNN)-based approaches have proved their ability to model robustly activity sets of various complexities, achieving state-of-the-art results [2]–[4]. However, DNNs require a large amount of data and tailored network designs to be able to robustly capture the differences between action classes. The DNN design complexity and training data requirements increases with the activity complexity, unavoidably leading to higher computational demands for complex activity sets.

Existing human activity recognition datasets can be grouped based on the activity complexity, into (a) coarse-grained, and, (b) fine-grained activity sets. Coarse-grained activities exhibit high inter- and low intra-class variations. In such cases, recognition is easier since critical features that differentiate the activity classes can be identified relatively easily, by focusing on specific appearance or motion characteristics. On the contrary, fine-grained activities involve complex actions composed by a set of sub-actions that may share similar characteristics in motion or appearance. This difference is also manifested in the size and complexity of the relevant action labels. In both cases, the vocabulary elements are selected to provide the best and shortest descriptions of the semantic content of the corresponding activities. Coarse-grained datasets achieve this with small-sized action label descriptions that have simple vocabulary content. On the other hand, fine-grained datasets involve actions with extensive descriptions and richer vocabulary content which is required to express the higher complexity of the corresponding activities.

In this work, we investigate the exploitation of information in the linguistic description of action labels, for HAR deep model design and learning. Our work contributes as follows:

- We present methods to identify and quantify possible similarities of action labels of different action classes.
- We show that these similarities, when used to define penalization weights, steer a DNN to learn finer representations for classes with similar linguistic descriptions.
- We highlight the existence of underlying relations be-

tween the parts of speech used for potential class similarity detection, and the type of the employed visual input.
- We evaluate the degree to which basic DNN architecture designs can benefit from this relation, and propose design guidelines for its best exploitation.

The remainder of the paper is organized as follows. We review related work on action recognition in Section II. Section III describes the proposed methodology. We evaluate our method in Sections V, IV and conclude in Section VI.

II. Related Work

In recent years, deep learning has dominated action recognition in videos, replacing hand-crafted spatio-temporal video representations. Its popularity is attributed to the automatic semantic representation learning of deep neural networks, that produces models with high discriminative capacity, without explicit definition of the model parameters as in hand-crafted descriptors.

Contrary to visual representation, the task of modeling the temporal structure of an action is still dominated by hand-crafted deterministic or probabilistic approaches that exploit either hand-crafted [5] or deeply-learned [6] feature descriptors. Regarding deep temporal structure modeling, the most common strategy is to define architectures that consist of recurrent neural networks [7] or 3-D convolutional kernels [8]. However, these DNN designs are not easily trainable [9] and require a large amount of data, which are lacking from existing action datasets. A widely adopted strategy by existing approaches in order to enrich the information quantity, is to define models that exploit multi-stream inputs that combine visual information with other, multi-modal information sources. Besides appearance and motion from visual data, these methods utilize language, audio or other sensory information to increase the representational power of their models.

Regarding language, the most prominent approach is to apply linguistic analysis on action script data with either hand-crafted [10], [11] or deeply-learned [12], [13] models and incorporate the additional knowledge to define more discriminative representations that are easier to model in action recognition or action captioning. The drawback of language-assisted approaches is that only a limited number of datasets provide script-based action descriptions, which restricts the applicability of these methods on a narrow set of action/activity cases. However, another source of linguistic information that is available in the majority of action datasets, are the action labels. They contain information about motion patterns (in the form of verbs) and the visual appearance of action-related aspects (in the form of nouns), such as objects. There exist only a few works that exploit a label-based linguistic analysis for action recognition and aim to learn verb-centered label correlations and transform the problem into multi-label classification [14], [15], jointly learning the two aspects. These methods focus on verb associations and require annotated data.

On the contrary, our work aims at applying label-based linguistic analysis for a-priori class similarity estimation with the goal of incorporating it to the learning process of deep architectures for action recognition in the form of misclassification penalization weights. The proposed approach aims to use Natural Language Processing (NLP) techniques to extract similarities between a set of action classes, by examining the lexical description of their class labels. This methodological direction stems from the observation that the semantics of class label descriptions contain information regarding possible associations between classes with respect to specific appearance or motion characteristics related to the input type. In detail, classes expected to contain similar appearance characteristics, will potentially have descriptions with similar syntax and vocabulary, consisting of words regarding the presence of objects used in the action(s) or general scene characteristics. On the contrary, classes with similar motion patterns are expected to have similar verb-centered lexical structures indicative of the performed action (e.g., “shoot a ball” or “shoot a basket”).

Our approach does not require any verb or noun-related annotations and no training because verb-relations are sourced from large semantic knowledge bases (in our case WordNet [16]) and refined using simple syntax rules based on a simple action description formulation.

III. Proposed Method

In the following, we present our approach on how the lexical analysis of action labels can identify their lexical similarities in order to derive the corresponding action class similarities. To this end, we present two distinct directions. The first has a local nature, focusing the analysis on a specific part-of-speech (verb, noun). The second utilizes the word semantics and ordering statistics to capture the global semantic context of the sentence. Subsequently, we demonstrate how to express these similarities in the form of weights. Finally, we show how to use these weights for training a neural network model and we demonstrate that by penalizing more severely the misclassification to a class with similar lexical descriptions enhances the activity recognition performance.

A. Part-of-speech weight generation

The initial approach on NLP-assisted weight class generation, relies on simple grammatical syntax, centered around a specific part-of-speech (e.g. verb, noun, subject). To define what can be considered as a simple syntax for an activity label sentence we formulate the following assumptions:

- Verbs characterize the motion motifs of the action and are expected at the beginning of the sentence.
- Nouns are expected to indicate the presence of an object that is either being used by the actor or is characteristic of the action, providing action-related appearance cues.
- Nouns are most likely to follow verbs.

According to these assumptions, we can define two weight generation strategies, (a) a verb-based approach, whose goal is to encode motion-related class similarities and (b) a noun-based approach which encodes the appearance-related ones.

*1) **Verb-based label processing**:* We begin by classifying each word into lexical categories such as nouns, verbs, adjectives, and adverbs. To achieve this we rely on a large set of corpus readers and lexical data resources provided by the Natural Language ToolKit (NLTK) platform [17]. Specifically, we used a part-of-speech tagger to process the sequences of words corresponding to each action class label in order to attach a part of speech tag to each word. Subsequently, we proceed by isolating the verbs that are placed in the beginning of the sentence. Additionally, we define simple syntax rules, to refine the set of unique verbs identified in the label set:

Syntax rule 1: *A verb can be followed by any number of particles (at, on, out, over per, that, up, with) or ad-positions (on, of, at, with, by, into, under). In such case, we define this candidate verb as compound.*

This rule stems from the requirement to distinguish between labels that share a common verb, but when followed by a particle or ad-position, can have different semantic interpretation (e.g., *take off* and *take out*).

After the verb-set refinement, the next step is to cluster together action classes with common verbs. In the case of non-compound verbs, a simple one-shot clustering is performed, grouping together classes that share the same verb. In the case of classes containing compound verbs, we also examine the degree of similarity between the accompanying particles (or ad-positions). For this approach, we compute the shortest path length between the word particles in the WordNet semantic knowledge base [16]. The path length is normalized in the range $[0, 1]$, with 1 indicating complete similarity. To accept two candidate classes as similar, despite verb sharing we also set an ad-position (or particle) similarity greater than 0.5.

Finally, we define a 2-D weight matrix, in which each row and column corresponds to an action class, and each cell (i, j) expresses whether the classes i and j are similar (value 1) or not (value 0). To transform the values to weights we normalize them so that, row-wise, the matrix entries sum to 1.

*2) **Noun-based label processing**:* Similarly to the previous approach, we start by classifying each word into lexical categories. Considering that a noun will most likely refer to an object/entity, its position most likely follows a verb with possible intermediate grammatical elements such as particles, ad-positions or delimiters (the, ., etc). This means that we must introduce complex syntax expressions to be able to accurately identify them. To achieve this, we exploit Noun Phrase chunking (NP-chunk) techniques [18] to segment and label multi-token sequences. In order to create an NP-chunker, we first define a chunk grammar, consisting of rules that indicate how sentences should be chunked. In our case, we rely on the simplistic expression of the action labels found the majority of human activity recognition datasets, to define a coarse grammar with a simple regular expression. This expression rule (syntax) states that chunks corresponding to noun candidates should follow the expression pattern, *"do something with something"*. Specifically, the syntax rule that we used to identify the nouns in the description has the following formulation:

Syntax rule 2: *A verb is followed by any number of particles or ad-positions, which is then followed by a delimiter, followed by a noun.*

Following the identification of candidate chunks and the isolation of unique nouns, we proceed as previously to cluster the input label sentences into the unique noun cluster centers. Finally, the associations between the classes that can be extracted based on the clustering result are used to generate the 2-D weight matrix, which expresses the degree of relation between the label sentences based on the present nouns.

B. Semantic Similarity-oriented weight generation

Relying the analysis on a specific part-of-speech leads to a key drawback. The prerequisite for correctly identifying similarities is that the class label sentences share common verbs or nouns. This aspect constrains the approach, since the simplified nature of the defined syntax rules does not adequately associate labels that consist of different verbs or nouns, but have identical semantic content. To alleviate this deficiency, we define a set of more complex rules based on the semantic context of sentences. To this end, we utilized the sentence semantic similarity assessment method of Yuhua et al. [19]. This approach extracts finer semantic similarity associations with the use of semantic and word order related metrics. The semantic similarity between two sentences is expressed by sourcing information about word relations from a structured lexical database and corpus statistics. Moreover, the exploitation of word order information enriches the semantic content expression, since word order affects the semantics of a sentence. In detail, the aforementioned semantic expression metrics are defined as follows.

Semantic similarity S_m: The semantic similarity S_m of two sentences is defined by first computing the *semantic vector* that characterize each sentence. The semantic vector of a sentence is derived based on the word set T_i it contains. Given two sentences and their word sets, the joint word set, $\mathbf{T}$, can be viewed as the semantic information on which their comparison is based. For each sentence i we compute its *lexical semantic vector* s_i based on T. The elements (corresponding to each word) of the semantic vector of a sentence i are defined based on the following criterion:

- If the word exists in both sets, then $s_i(w_k)$ is set to 1, where w_{k_i} is the k-th word of the sentence i.
- Otherwise, $s_i(w_k)$ is defined by considering the set of synonyms (synset) of w_k and a word of sentence j, the path lengths connecting w_{k_i} to w_{m_j}, and the relative depth of the common ancestor word of w_{k_i} and w_{m_j} in WordNet.

In more detail, for the latter case, the semantic vector value depends on synset overlapping cases which, briefly explained, are defined as:

- *Mutual synset*: For words belonging to the same synset, i.e. w_{k_i} belongs to the synset of w_{m_j} and vice versa, $s_i(w_k)$ is set to 1.

- *Indirect synset element sharing*: For words not belonging to the same synset, but whose synsets have common words, the $s_i(w_k)$ is set to the constant e.
- *Discrete synsets*: For words that neither belong to the same synset, nor have synsets containing common words, the value of $s_i(w_k)$ is defined as a the product of (a) the exponential function of the shortest path length between w_{k_i} and w_{mj}, and (b) the hyperbolic tangent of the relative depth of the common ancestor word. The second function expresses the analogous increase of depth and semantic similarity between words. This means that for this case we need to search for the common ancestor with the most relative semantic context to both words [1].

Finally, the semantic similarity S_m between two sentences is defined as the cosine coefficient between their corresponding lexical semantic vectors.

Word ordering S_o: We assign a unique index number for each word in each sentence's word set, T_i which refers to the order in which the word appears in the sentence. To compute the word order similarity, a *word order vector* r_i, is formed for each sentence's word set, based on the joint word set T. The r_i estimation, for a sentence i, involves the following cases:

- For words present in both sentences i and j, r_i is filled with the word index.
- For words not present in both sentences, the most similar word in T_j is found based on the synset inclusion. If the similarity between the initial word and the most similar word is greater than a preset threshold (set to 0.2 in [19]), r_i is filled with the index number of the similar word in T_j. Otherwise, r_i is filled with zero for that word.

The word ordering similarity S_o between the two sentences, described by the their *word order vectors*, r_1 and r_2, is expressed as:

$$S_o = 1 - \frac{|r_1 - r_2|}{|r_1 + r_2|}. \quad (1)$$

Eventually, the overall sentence similarity is defined as a combination of semantic similarity and word order similarity:

$$S_{tot} = \delta \cdot S_m + (1 - \delta) \cdot S_o \quad (2)$$

where δ is a user-defined scalar controlling the influence of each metric to the overall similarity score. Based on the original paper guidelines, we set δ to 0.7, favoring semantic similarity over syntax.

The final step is to use S_{tot} to compute a normalized 2-D weight matrix which expresses the degree of relation between the label sentences based on overall sentence similarity.

C. Weighted classification loss

One way of exploiting prior knowledge about class similarity derived by the lexical analysis of their action labels is through the definition of penalization weights that can be used in the loss function guiding the learning process of a neural network architecture designed for the task of human activity recognition. The majority of existing works that apply deep learning in human activity recognition, rely on the regular categorical cross-entropy loss function. In our case, however, we deviate from this trend and resort to the use of focal loss [20]. The selection of focal loss stems from the fact that in complex fine-grained activity datasets, such as the MPII Cooking dataset, extreme class imbalance is present. In such cases, as the authors argue, it is more logical to down-weight the loss assigned to well-classified examples in order to again focus on the misclassification cases. With the incorporation of penalization weights the loss function is defined as:

$$\mathbf{L}_{loss} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K}(w_k + 1)\,(1 - Y_{n,k})^{\gamma}\, log\,(Y_{n,k})\,, \quad (3)$$

where N is the number of training samples, K the number of action classes, Y the estimated label, and w a vector of similarity weights for each class compared to the rest, that is generated from the employed NLP method. The loss function contains a modulating factor $(1 - Y_n)^{\gamma}$ to the cross entropy loss, with tunable focusing parameter $\gamma \geq 0$. When γ equals to 0, the loss is equivalent to the classic cross-entropy categorical loss. As γ increases, the effect of the modulating factor is enhanced. In [20] the authors verified experimentally that when $\gamma = 2$, the loss achieved its full potential. We also maintained this configuration in all of our experiments.

D. Extension to multi-stream NN architectures

Equation (3) can be extended to multi-stream neural architectures, thus keeping up with the neural network design directions of recent state-of-the-art methods. In fact, as we present in the following section, it is more beneficial to assign to each sub-network a specific input stream-related weight generation strategy, and even try to reproduce the levels of semantic complexity (from simple part-of-speech up to semantic context similarity) in the deep neural network architecture design.

IV. Experimental set-up

For the experimental assessment of the proposed weight generation schemes we defined both single and multi-stream (two-stream) architectures, using a custom, baseline spatio-temporal neural network design based on Long-Short Memory (LSTM) cells, Fully-Connected (Dense) and Convolutional layers. The organization of the experiments and the presentation of the results are intended to (a) highlight the performance gain when the proposed NLP-based weight generation process is applied to both single and multi-stream DNN architectures, (b) present the existence of a correlation between the selection of the part-of-speech used in the label similarity assessment and the type of input information used, (c) demonstrate the way we can exploit this correlation in a multi-stream DNN design, by also introducing the concept of semantic complexity to the design, and, (d) show that the contribution of the generated weights is reinforced if we constrain the region of action introduced in the DNN architecture to the region of the actor performing the action and the object being manipulated.

[1]For more details the interested reader is referred to the original paper [19]

A. Neural network architecture

In the context of evaluating our proposal we designed a single stream spatio-temporal action recognition neural network architecture, which is easily modifiable into a multi-stream design. The architecture consists of two sub-networks that follow the standard modeling structure of an action sequence recognition pipeline, with the first being responsible for the frame-wise feature extraction task and the second for the temporal modeling of the feature sequence and the classification. In order to evaluate possible correlations between the selection of part of the speech and the type of input, we evaluate the architecture (s) in the use of RGB and color-encoded optical flow frame data.

Feature extraction: Instead of designing a new spatial modeling network, we opted to utilize the widely used VGG-16 [2] network. From this network we extract 2-D feature vectors from the last 2-D layer, resulting in a feature tensor of 7-by-7-by-512 for each frame of the sequence, as well as the 1-by-2048-dimensional feature vector from the last fully-connected layer. In the case of RGB frames we do not fine-tune the network and maintain the learned weights from the training on ImageNet [21]. For optical flow data we fine-tune the VGG-16 layers starting from the last 2-D layer and above on optical flow data from the KTH dataset [22], freezing the rest with the weight values from ImageNet.

Temporal modeling: The sub-network for this task consists of two Bidirectional Long Short-Term Memory (BiLSTM) layers, followed by three fully-connected (FC/Dense) layers with Leaky ReLU and soft-max activation functions. Moreover, between the first and second FC layers we introduce batch normalization, and between the second and third, dropout with 0.7 unit rejection fraction rate. For the sake of the execution speed of the performed experiments, and to further evaluate the recognition capacity of the designed temporal architecture, we defined two variations based on the input dimensionality:

- *1-D feature sequences*: If the frame-wise input is a 1-by-2048 feature vector, we use BiLSTM cells.
- *2-D feature sequences*: If the frame-wise input is a 7-by-7-by-512 feature map, we alternate the BiLSTM cells with 2-D Convolutional BiLSTM cells.

We expect the second variant to have better overall performance since the spatial relations between neighboring regions are maintained, enriching its representational expressiveness.

B. Datasets

To evaluate the performance of the proposed weight generation strategies we employed three widely used human activity recognition datasets, namely the MHAD [23], the J-HMDB [24] and the MPII Cooking Activities [25] datasets. The datasets are exemplar cases of coarse-grained, mid-range, and, fine-grained activity sets. Additionally, in these datasets the action complexity is reflected by the size and complexity of the action label descriptions. Table I presents statistics based on the size, number of unique verbs/nouns as well as the average number of classes to which a class is related to and the minimum/maximum number of relevant associations.

TABLE I
DATASET LABEL STATISTICS

	Datasets		
	MHAD	***J-HMDB***	***MPII Cooking***
Num unique verbs	9 verbs	19 verbs	42 verbs
Num unique nouns	5 nouns	6 nouns	28 nouns
Avg num verbs/lbl	1.128 verb/lbl	1.0 verb/lbl	1.188 verbs/lbl
Avg num nouns/lbl	0.700 noun/lbl	0.333 noun/lbl	0.610 nouns/lbl
Avg lbl length	3.182 PoS/lbl	1.333 PoS/lbl	2.297 PoS/lbl
Avg asc via verb	0.545 asc/lbl	0.286 asc/lbl	1.656 asc/lbl
Avg asc via noun	0.818* asc/lbl	0.191 asc/lbl	0.844 asc/lbl
Max/min asc verb	1/0 asc	2/0 asc	5/0 asc
Max/min asc noun	3/0* asc	1/0 asc	3/0 asc

Abbreviations utilized in the table contents are the following, *Avg*: average, *num*: number, *PoS*: part-of-speech, *lbl*: label, *Average asc*: average number of classes a single class is related to based on a part-of-speech, *asc*: associations, refers to the amount of class label lexical relations based on a specific PoS. *: In this dataset we also include the word *hand(s)* as noun despite the fact it refers to a human body part rather than an object.

Moreover, these datasets cover a wide range of action characteristics. MHAD and MPII consist of action sequences in a constrained environment, whereas J-HMDB (which is a sub-set of HMDB dataset [26]), consists of action sequences in the wild, with videos taken from YouTube. MHAD consists of 11 action classes with no large scene and actor appearance variation and without human-object interactions. J-HMDB consists of 21 classes, with large variations in the scene appearance containing human-object interaction. MPII Cooking is the most challenging of the three datasets, consisting of 64 classes, with low inter-class and high intra-class similarities, and human-object interactions. In general, these three datasets are quite diverse with respect to the number of action classes, the complexity of the actions and the intra-class action similarities.

In the experimental part, we follow the standard experimental protocol described in the corresponding baseline dataset papers and report, for the case of multiple splits, the average accuracy across all splits.

V. EXPERIMENTAL RESULTS

We now present the results of the experiments presented in section IV. The batch size for MPII and J-HMDB was set to 72 samples per batch, whereas for MHAD to 64. The networks were trained for 20K iterations for MHAD and J-HMDB and for 38K iterations for the MPII dataset, in a Nvidia Quadro P6000 GPU. The loss minimization is performed using the Adadelta optimizer. For MPII Cooking, due to the large range of action segment sizes, we train the networks with video clips of 10 frames, sampled uniformly across the entire sequence.

A. Linguistic analysis complexity in relation to input type and activity complexity

The first set of experiments aims at investigating the correlation between the complexity of the natural language processing approach (part-of-speech or semantic sentence similarity), the input type (appearance-RGB, motion-Optical

TABLE II
INPUT SOURCE AND WEIGHT STRATEGY RELATION

Architecture Design	Datasets		
	MHAD	*J-HMDB*	*MPII Cooking*
RGB - BiLSTM (baseline)	63.19%	38.81%	30.69%
RGB - BiLSTM & WV	67.81%	38.92%	30.25%
RGB - BiLSTM & WN	**69.20**%	**40.35**%	32.48%
RGB - BiLSTM & WSM	66.04%	37.69%	**33.10**%
OF - BiLSTM (baseline)	72.40%	38.33%	31.87%
OF - BiLSTM & WV	**75.14**%	**40.88**%	33.55%
OF - BiLSTM & WN	73.55%	38.19%	32.36%
OF - BiLSTM & WSM	74.32%	35.81%	**34.19**%

Mean accuracy for each dataset. We refer to the weight generation strategies as follows. Baseline: without weighting, WV: verb-based, WN: noun-based, WSM: semantic similarity-based.

Flow) and the complexity and degree of similarity between the actions/activities. Table II presents our experimental results regarding all possible relations between the weight generation strategy and the input source. To evaluate this, we use the single-stream temporal architecture with 1-D input feature vectors only, i.e., we do not consider at this point the 2-D variant. This is entirely due to issues of computational performance, learning cost, and the fact that the incorporation of weights to the learning process is independent of the input dimensionality. In Table II we can observe that there exists a direct correlation between the input type and the goal of the lexical analysis. In the case of optical flow data, which imprints the motion information in an action, the application of a verb-based lexical analysis and weight generation is more beneficial since actions with similar motion motifs will most likely be described by a similar verb. Analogously, for RGB image data which encode the appearance information of the scene, the adoption of a noun-based direction leads to better results, since actions containing the same object-in-use will have similar appearance characteristics. Overall, the introduction of NLP-driven weights into the learning process of a neural network architecture is beneficial, as long as the label annotation provides meaningful descriptions.

Regarding the underlying association between activity, label sentence complexity and lexical analysis direction, we can observe that in dataset cases with simplistic and small-length label sentences, adopting an analysis focused in a specific part-of-speech instead on a global semantic level is better. In contrast, for datasets with larger and more complex action labels with larger semantic context (such as in MPII) the contribution of the semantic similarity weight generation strategy is greater. This can be attributed to the fact that this natural language processing method is able to consider the semantic context of the sentences compared to the token-focused methods (verb, noun) that are not able to encapsulate semantic context due to their simplified structure.

In any case, the lexical analysis of any kind appears to improve the accuracy obtained by the baseline networks that do not involve it.

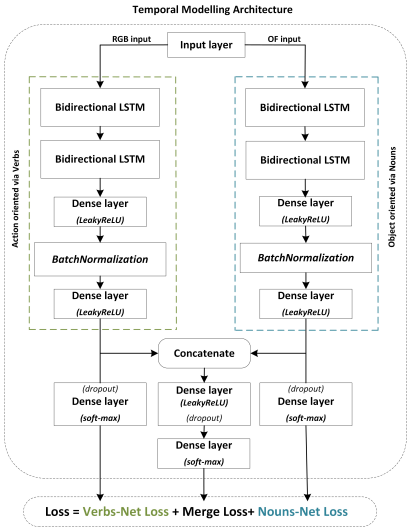

Fig. 1. The employed temporal modeling architecture, a two-stream BiLSTM cell-based design. Last layers are fused together via concatenation. The fused feature vectors are processed with a set of fully connected layers. The minimization loss is three-fold. The motion-related loss is weighted using a verb-based approach, the appearance-related loss with noun-based weights and the merge loss is weighted using the semantic similarity-based approach.

B. DNN design complexity and weight applicability

We also investigate the relationship between the number of input sources and sub-networks in the temporal NN design, with the NLP-assisted weight penalization method applied to each of them depending on the type of input, taking advantage of the previous findings. For this set of experiments we compare a single-stream (RGB or optical flow) deep architecture to a two-stream (RGB, optical flow), whose abstract design specifications were provided in Section IV. For the two-stream design, based on the previous findings, we apply verb-based weights to the sub-network introduced with optical flow data, and noun-based weights to the RGB-induced sub-net. Moreover, following the literature guidelines [27], we fuse the representations of the two sub-networks via concatenation and forward the outcome to a set of two fully-connected layers that produce the final classification. An illustration of the two-stream DNN architecture is provided in Figure 1. This design direction reforms the minimization loss into:

$$\mathbf{L}_{total} = L_{appear} + L_{motion} + L_{fused}. \quad (4)$$

The previous formulation resembles the hierarchical structure of a linguistic sentence describing an action. By concatenating the appearance and motion representations we attempt to encode the semantic context of the action. To this end, we examine whether the application of class similarity weights reflecting semantic information of the same level generated from the label sentence, can further assist the learning process. Finally, we examine the benefit of using 2-D features as network inputs, instead of 1-D feature vectors. The results of these set of experiments are presented in Table III. The obtained experimental results verify the modeling approach followed by the research community which states that the

TABLE III
RELATION BETWEEN ARCHITECTURE COMPLEXITY AND NLP-ASSISTED WEIGHT INCORPORATION INTO LEARNING.

Architecture Design	Datasets		
	MHAD	***J-HMDB***	***MPII***
RGB, BiLSTM & *WN*	69.20%	40.35%	32.48%
OF, BiLSTM & *WV*	75.14%	40.88%	33.55%
Both, BiLSTM, (*WWL,WWL,WWL*)	77.57%	39.84%	34.12%
Both, BiLSTM, (*WN,WV,WWL*)	79.08%	42.36%	36.78%
Both, BiLSTM, (*WN,WV,WSM*)	82.57%	47.10%	39.46%
Both, BiLSTM2D, (*WN,WV,WSM*)	85.13%	49.89%	41.50%

Mean accuracy for each dataset. We refer to the weighting strategies of the three architecture sub-networks as WWL: without weight learning, WV: verb-based, WN: noun-based weight generation, WSM: semantic similarity-based. The rest of the abbreviations are BiLSTM: Bidirectional Long Short-Term Memory, BiLSTM2D: Bidirectional Convolutional Long Short-Term Memory with 2-D inputs.

combined motion and appearance-related information leads to an accuracy increase. In fact, regarding the non-weighted deep architectures, the improvement reached up to 5%.

The NLP-assisted weight generation and the incorporation of such a policy into the learning process of DNNs appears to be beneficial for both single-stream and multi-stream DNN architectures, with an additional improvement in the range of 1.5% to 2.5%, with the rate of increase being inversely proportional to the complexity of the action and the features of the dataset. Moreover, the adoption of a multi-level weighting strategy that reflects the different levels of interpretation of the action label semantics and its application to a multi-stream DNN architecture, assists the learning process, leading to an overall accuracy increase in the range of 4% to 7%, compared to a non-weighted multi-stream design.

C. Focusing on the actor and on the object-in-use

The purpose of the last set of experiments is to examine the impact of refining the region of interest with the scope of strengthening the underlying relation between the input source characteristics and the linguistic description of the action in the label sentence. The factors that contribute more significantly in recognizing the action in an video sequence are the actor's and the manipulated object's characteristics (motion or appearance). Restricting the region of interest to the region(s) of these contributors is considered a standard approach in recent methods, increasing performance since noisy/irrelevant information is removed.

In a linguistic description of an action (such as the action label) the object-in-use is most likely included as a noun (e.g., throw a ball). In the following experiments we also investigate whether the application of noun-based wight generation indirectly requires the presence of the object appearance and motion information in order to be effective.

To detect the actor and object-in-use regions of interest we used Mask-RCNN [28]. We did not define and retrain Mask-RCNN to the individual dataset and the object classes they contain, but we rather used the pre-trained Mask-RCNN weights obtained on the COCO [29] dataset. This choice is justified by the fact that the majority of the objects in the employed datasets are included in COCO. To distinguish the object most likely in use by the actor from the set of other detected objects, we consider the Euclidean distance between the centroid of each candidate object and the centroid of the actor's hand. The hand centroid is generated based on the 2-D locations of hand key-points, extracted using OpenPose [30].

TABLE IV
ACTOR, OBJECT-IN-USE IMPACT

Architecture Design	Datasets		
	MHAD	***J-HMDB***	***MPII***
FF, BiLSTM, (*WN,WV,WSM*)	82.57%	47.10%	39.46%
A, BiLSTM, (*WN,WV,WSM*)	84.40%	46.55%	37.72%
AO, BiLSTM, (*WN,WV,WSM*)	86.19%	50.02%	43.85%
AO, BiLSTM2D, (*WN,WV,WSM*)	90.38%	52.09%	45.65%

Mean accuracy for each dataset. New abbreviations in this table are FF: Full frame as input A: Actor region as input, AO: Actor and Object regions as inputs, with the corresponding feature vectors fused via concatenation. Both appearance and optical flow features are used in all cases.

Table IV presents the accuracy variation due to the introduction of the actor-object detection, and the indirect relation between noun-based weighting and object presence. Regarding MHAD, which is a coarse-grained action dataset without human object interactions, we verified that the detected and selected object remains the same for all action cases. It can be verified that restricting the action modeling to the regions containing the actor and the manipulated object provides more informative cues regarding the performed action. Moreover, as expected, preserving the spatial (appearance and motion-based) characteristics of both actor and object is beneficial for the majority of methods. Even more, combining this strategy with the proposed weight generation scheme, improves further the recognition capabilities of the architecture.

Regarding indirect relations between the actor, object detection and the weighting strategy, the results indicate that the contribution of noun-centered weights on the appearance sub-network is higher when object-related information accompanies the actor-related. This indicates an underlying association between the visual and linguistic representation of objects.

Overall, we can observe that human-object interaction action datasets appear to benefit more from an action label-oriented similarity weight penalization, compared to datasets that do not exhibit this characteristic.

VI. CONCLUSIONS AND FUTURE WORK

This paper investigated the applicability of natural language processing on action label sentences to derive potential action class similarities. We demonstrated that these derivations, when formulated as misclassification penalization weights, can be incorporated in the learning process of DNN architectures to assist and enhance the modeling capacity and the classification accuracy of such architectures. The experimental results indicate that the incorporation of this a priori knowledge increases performance, by guiding the network to learn finer representations for actions that share similar linguistic

descriptions. Moreover, it has been shown that there exists an indirect association between the input type and the selected tokenization strategy. Specifically, appearance-based input types benefit from noun-based similarity extraction whereas motion-based input types from a verb-based similarity extraction. Even more, this formulation is applicable to any deep learning action recognition architecture, without the requirement of intervening on the architecture design.

However, this work also revealed additional requirements and limitations of this strategy. A requirement for extracting meaningful and correct class associations is the existence of informative label annotations in the form of action verbs and nouns for objects, for human-object interaction activity cases. Regarding the choice between a local, token-specific and a global, semantic context encoding analysis, it is evident that the criterion lies on the size and lexical complexity of the action label. Yet, further investigation is required to define the boundaries, parameters and refine the natural language processing methods so as to distill more accurately the underlying associations between the action classes based on their labels. It is also evident that the information of the semantic context of label sentences introduces additional knowledge regarding the formulation and complexity of the action/activity. This could inspire new designs that exploit the different levels of semantic relations to produce finer and clearer representations, with the potential to achieve higher recognition scores. Ongoing research is focused in that direction.

ACKNOWLEDGMENT

This work was supported by the Hellenic Foundation for Research and Innovation (HFRI) under the HFRI PhD Fellowship grant (Fellowship Number: 1516) and the European Community via the project Co4Robots (H2020-731869). We also acknowledge the support of NVIDIA Corporation for the donation of a GPU.

REFERENCES

[1] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, p. 16, 2011.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] G. Gkioxari, R. Girshick, and J. Malik, "Contextual action recognition with r* cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1080–1088.

[4] M. U. Khalid and J. Yu, "Multi-modal three-stream network for action recognition," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 3210–3215.

[5] J. Lei, X. Ren, and D. Fox, "Fine-grained kitchen activity recognition using rgb-d," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012, pp. 208–211.

[6] C. Lea, A. Reiter, R. Vidal, and G. D. Hager, "Segmental spatiotemporal cnns for fine-grained action segmentation," in *European Conference on Computer Vision*. Springer, 2016, pp. 36–52.

[7] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[8] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[9] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, 2013, pp. 1310–1318.

[10] M. Rohrbach, A. Rohrbach, M. Regneri, S. Amin, M. Andriluka, M. Pinkal, and B. Schiele, "Recognizing fine-grained and composite activities using hand-centric features and script data," *International Journal of Computer Vision*, vol. 119, no. 3, pp. 346–373, 2016.

[11] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," 2008.

[12] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," in *European Conference on Computer Vision*. Springer, 2016, pp. 852–869.

[13] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.

[14] S. Khamis and L. S. Davis, "Walking and talking: A bilinear approach to multi-label action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 1–8.

[15] M. Wray and D. Damen, "Learning visual actions using multiple verb-only labels," *arXiv preprint arXiv:1907.11117*, 2019.

[16] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, vol. 38, no. 11, p. 39–41, Nov. 1995. [Online]. Available: https://doi.org/10.1145/219717.219748

[17] E. Loper and S. Bird, "Nltk: The natural language toolkit," in *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ser. ETMTNLP '02. USA: Association for Computational Linguistics, 2002, p. 63–70. [Online]. Available: https://doi.org/10.3115/1118108.1118117

[18] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media, 2009.

[19] Y. Li, D. McLean, Z. A. Bandar, K. Crockett *et al.*, "Sentence similarity based on semantic nets and corpus statistics," *IEEE Transactions on Knowledge & Data Engineering*, no. 8, pp. 1138–1150, 2006.

[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[22] I. Laptev, B. Caputo *et al.*, "Recognizing human actions: a local svm approach," in *null*. IEEE, 2004, pp. 32–36.

[23] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Berkeley mhad: A comprehensive multimodal human action database," in *2013 IEEE Workshop on Applications of Computer Vision (WACV)*. IEEE, 2013, pp. 53–60.

[24] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3192–3199.

[25] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele, "A database for fine grained activity detection of cooking activities," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1194–1201.

[26] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[27] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1933–1941.

[28] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask r-cnn," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

[29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[30] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7291–7299.